ADAPTIVE IMAGE FILTERING BASED ON A DISTANCE TRANSFORM

BACKGROUND

1. Field

This disclosure relates to compression of electronic image signals, more particularly to removal of distortions in reconstructing of these compressed image signals.

2. Background

Block-based DCT (Discrete Cosine Transform) coding has been used in many image compression standards, such as H.261, H.263, JPEG, MPEG-1, and MPEG-2. This is a lossy compression technique, meaning that there are variations in the output image when compared to the original image. The compression process causes visual artifacts in the decoded images, two of which are referred to as blocking and ringing. Blocking artifacts occur along the block boundaries and ringing artifacts appear near strong image edges, both due to the coarse quantization of the DCT coefficients.

Among the coding artifacts, ringing is the most visible one in high quality systems, those having low compression ratios. One typical example of the ringing artifacts is the noise surrounding strong image edges that represent text and graphical object boundaries. In video coding, due to slight variations from frame to frame, ringing noise is visible in moving pictures as a local flickering near edges. This type of noise is known as "mosquito noise."

In general, for a given compression ratio, there is a limitation on the visual quality of an image. Quantization tables can be carefully designed to get higher quality images, however, the tradeoff is to get lower compression ratio. Therefore, post-processing techniques have been introduced to remove the image artifacts from the reconstructed images. A postprocessor can process a reconstructed image signal to reduce or remove the image artifacts resulting from quantization distortions. The general rule of thumb is that image edges should be preserved while rest of the image should be smoothed with spatial filters carefully chosen based on the characteristic of a particular pixel or set of surrounding pixels.

Most of the current post processing techniques have focused on reducing the artifacts produced by DCT coding, in particular, reducing the blocking artifacts. These techniques operate along block boundaries, and have difficulty in removing the ringing artifact that extends over a larger area. Some examples of these types of techniques can be found in US

Patent Nos. 5,850,294, issued December 15, 1998; 5,512,956, issued April 30, 1996; and 5,473,384, issued December 5, 1995.

The '294 patent, for example, uses one-dimensional lowpass filters along the block boundaries to reduce visual artifacts, including blocking and ringing, in DCT coded images. The '956 patent performs three-dimensional post processing to reduce artifacts produced by block-based motion-compensated coding. Spatial and temporal filtering is applied and then a one-dimensional median filter is used in the spatial domain. The '384 patent provides a description of a post-filtering method with a temporal filter and a spatial filter. A multilevel median filter is applied in the spatial domain.

The techniques of the '384 patent can also remove certain amount of ringing noises. Most current techniques in this area, including the approach in the '384 patent have high computational complexity that makes them inefficient for many applications. Other techniques of this type can be found in US Patent No. 5,920,356, issued July 6, 1999; and US Patent No. 5,819,035, issued October 6, 1998. The '035 patent applies anisotropic diffusion on coded images to reduce ringing artifacts. Lowpass filtering is done by an iterative diffusion process. The '356 patent applies three-by-three lowpass filters in the spatial domain, and uses both temporal and edge characteristics of the video image to enhance the displayed image.

As mentioned above, these techniques have computation complexity and makes them inefficient for many applications. Also, most of the above techniques are directed specifically to DCT-based coding. Therefore, a more general purpose, efficient and simple technique for removing ringing artifacts from images reconstructed from any compression scheme would be useful.

SUMMARY

One aspect of the disclosure is a method for reducing visual artifacts in reconstructed images. The method determines edge energy for each pixel in the image and then compares the edge energy for each pixel to a threshold, producing an edge map. A distance transform is then used to produce a filter map and a filter is applied to pixel in the image, such that the filter applied is dependent upon a filter map value for each pixel. An output value for each pixel is then produced.

Another aspect of the disclosure is a facsimile machine that performs adaptive filtering on a decompressed image to remove visual artifacts.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention may be best understood by reading the disclosure with reference to the drawings, wherein:

Figure 1 shows an image broken up into image blocks for post processing, in accordance with the invention.

Figure 2 shows a flowchart of one embodiment of a method for post processing reconstructed image signals, in accordance with the invention.

Figure 3 shows horizontal and vertical Sobel filters.

Figure 4 shows one embodiment of a system in which post processing is performed on reconstructed image signals, in accordance with the invention.

DETAILED DESCRIPTION OF THE EMBODIMENTS

In one embodiment, the postprocessor processes images in a block-wise style, which means that the postprocessor processes one image block at each time. This approach requires very limited memory space in hardware implementation, independent from the size of the image being processed. An example is shown in Figure 1, with an image 10 broken into image blocks such as 12. For instance, the image blocks 12 can be arranged horizontally across each strip of a large image.

$B_{m,n}(i,j)$, with $i = 0, \ldots\ldots, (M - 1)$, and $j = 0, \ldots\ldots, (N - 1)$, is used to represent an M-by-N image block, where the coordinate of pixel (i,j) in the whole image is

$$B_{m,n}(i,j) = (mu + i, nv + j)$$

The variables u and v represent the processing step size; while M and N are defined as $(u + p)$ and $(v + p)$, respectively; and p is a padding parameter. The padding parameter is necessary because of the convolution operations applied in the postprocessor; and its value relies on the specific convolution operations. In one embodiment, assume that $u = v = 32$, in favor of future hardware development. In one embodiment, set p=4 when 3-by-3 convolution kernels are applied.

A method of adaptively filtering a reconstructed image is shown in Figure 2. In one embodiment of a postprocessor, different color components, such as R, G, and B, or L, A, and B, as examples, of an image are processed independently. The post processing for each color component of each image block starts with edge detection at 20. Edge filters, such as Sobel

filters, Prewitt filters, and DOG (Derivative of Gaussian) filters, can be used to generate the horizontal and vertical edge energies (E_h, E_v) for each image pixel. One embodiment used Sobel filters, such as those shown in Figure 3.

The total edge energy for each pixel is then taken as

$$E(i,j) = |E_h(i,j)| + |E_v(i,j)|$$

where E_h is a summation of all of the pixel values of the surrounding pixels multiplied by the corresponding term in the horizontal filter with the current pixel *p(i, j)* located at the enter position of the filter. A similar calculation is performed for E_v using the vertical filter.

An *M*-by-*N* edge map *P* is then derived at 22 from the edge energy by thresholding.

$$P(i,j) = \begin{cases} 1 & E(i,j) > T \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

The pixels with edge energy above the threshold *T* are labeled as edge pixels and set to 1 in the edge map.

Ringing artifacts usually occur on flat backgrounds near strong edges. The artifacts are stronger than the background but weaker than the edge. Therefore, if the local edge strength is known, it can be used to define a threshold below which a variation is insignificant. The threshold *T* can be defined as the following

$$T = \min(T_0, E_{\max}/2) \quad (4)$$

where T_0 is a user-defined maximum threshold, $E_{\max}$ is the maximum edge energy within the current image block, which means that the threshold *T* can be different for different image blocks. Based on experiment, T_0 is set to 184 in this particular embodiment of a postprocessor.

To avoid introducing artifacts across the image block boundaries during post processing, the edge map boundary pixels are set as edge pixels, i.e.,

$$P(i,j) = 1 \quad \text{if } i < 2 \text{ or } i > (M\text{-}3) \text{ or } j < 2 \text{ or } j > (N\text{-}3) \quad (5)$$

In one embodiment, when *T* is less than a small value, such as 2 or 4, the process skips the post processing steps completely. This is shown in Figure 2, where the threshold, T, is compared to a predetermined minimum value MIN. If T is less than MIN, the process ends. This approach effectively skips the post processing for a smooth block. It can significantly reduce the computational time when large areas of smooth background exist in an image.

From the *M*-by-*N* edge map *P*, a *M*-by-*N* filter map $F(i,j)$ is generated at 24 using a distance transform, which is defined as the following.

$$F(i,j) = \min(D, \text{distance from pixel } (i,j) \text{ to the nearest edge pixel in the edge map}) \quad (6)$$

where D is the user-defined maximum distance. For instance, the edge pixels labeled in the edge map are given a value of 0 in the filter map. In one embodiment, the value of D is set to 3, in favor of most current microprocessors. In one embodiment, the distance between two pixels is defined as the maxim of the horizontal and the vertical coordinate differences of the two pixels.

Low-pass filters are then chosen adaptively for each image pixel according to the filter map F. For pixel (i,j), the size of the filter is $(2F(i,j)+1)$-by-$(2F(i,j)+1)$. To reduce the computational burden, normalized structure elements are used as filters in one embodiment of a postprocessor. A normalized structure element means a normalized filter, where coefficients are all equal. Low-pass filters other than the normalized structure elements can also be applied during the adaptive filtering 26. To further reduce the computational complexity, the 2D filtering process can be performed in two steps using a horizontal and then a vertical 1D filters. In order to keep sharp image edges, for each image pixel, the filter element is set to 0 if corresponding neighboring pixel is shown as edge pixel in the edge map.

An alternative way for the two-step 1D filtering process is to use separate filter maps for horizontal and vertical operations, respectively. The filter maps are defined as the following.

$$F_h(i,j) = \min(D, \text{distance from pixel } (i,j) \text{ to the horizontally nearest edge pixel in the edge map})$$

$$F_v(i,j) = \min(D, \text{distance from pixel } (i,j) \text{ to the vertically nearest edge pixel in the edge map})$$

The filter size is $(2F_h(i,j)+1)$ and $(2F_v(i,j)+1)$ for horizontal and vertical filters, respectively. Since only 1D filter maps are derived and applied during post processing, this approach can further speed up the computation.

As mentioned, in order to keep sharp image edges, for each image pixel, the filter element is set to 0 if corresponding neighboring pixel is shown as edge pixel in the edge map. This means branch operations are heavily used, which increases the computational complexity. To avoid the branch operations and significantly reduce the computational time, the filter size can be changed to $(2F_h(i,j)-1)$ and $(2F_v(i,j)-1)$ for 1D filters and $(2F(i,j)-1)$-by-$(2F(i,j)-1)$ for 2D filters. The tradeoff is the degradation in the post-processing quality.

One of the potential applications of the adaptive image filtering technique is: post processing for images with texts and graphics. A more specific example is JPEG-based Color

Fax. The postprocessor can be used in a color fax machine 30 to clean up the reconstructed image and deliver high-quality output, as shown in Figure 4.

A port 32 is operable to receive a compressed image. The compressed image is then passed to a processor 34 that decompresses the image. Post-processor 38 then performs adaptive filtering as set out above, to remove the objectionable visual artifacts. This decompressed and filtered image is then passed to the print engine 36 for production as an output document. This output document may be a color document.

In some embodiments, the processor 34 and the post-processor 38 may be included in the same processor, but embodied by the processor executing different instructions when it acts as a processor and as a post-processor. Further, the methods of the invention may be performed by instructions contained on an article. The instructions, when executed, perform the methods of the invention.

The artifact reducing technique described here is applicable in many image (and video) processing systems, which employ block-based DCT coding schemes, such as JPEG, MPEG-1, MPEG-2, MPEG-4, H.261, and H.263. Since no DCT or any other coding information is considered in the postprocessor described here, the method can basically be used for image enhancement with any image coding schemes.

Values of parameters, such as the user-defined maximum distance D, maximum edge energy threshold T_0, image block width M and height N, provide flexibility for different applications.

Thus, although there has been described to this point a particular embodiment for adaptively filtering reconstructed image data, it is not intended that such specific references be considered as limitations upon the scope of this invention except in-so-far as set forth in the following claims.